

On Exploiting Energy-Aware Scheduling Algorithms for MDE-based Design Space Exploration of MP2SoC

Manel Ammar and Mouna Baklouti
CES Laboratory
National Engineering School of Sfax
Sfax, Tunisia
Email: manel.ammar@ceslab.org

Maxime Pelcat and Karol Desnos
IETR, INSA Rennes
CNRS UMR 6164, UEB
Rennes, France
Email: mpelcat, kdesnos@insa-rennes.fr

Mohamed Abid
CES Laboratory
National Engineering School of Sfax
Sfax, Tunisia

Abstract—Massively Parallel Multi-Processors System-on-Chip (MP2SoC) architectures have been widely deployed to run challenging high-performance computations. However, the ever greater demand for energy efficiency fosters energy budgeting in MP2SoC systems. Nowadays, having the appropriate Electronic Design Automation (EDA) tools for power estimation is mandatory. The major challenge for the design of such tools is to reach a better tradeoff between accuracy and time-to-market. This paper presents a Model Driven Engineering (MDE)-based energy-aware Design Space Exploration (DSE) approach allowing the designer to take the power consumption criterion into account early in the design flow. The originality of this approach is that it integrates the Energy-Aware Duplication (EAD) algorithm that strives to balance schedule lengths and energy savings by considering the most important sources of energy consumption in MP2SoC: the massive number of processing elements (PE) and the high-speed Network-on-Chip (NoC). To demonstrate the effectiveness of the proposed approach, we conducted experiments using the H.263 encoder application. The obtained results demonstrated that EAD can effectively save energy in MP2SoC systems. They also showed that our MDE approach is capable of accelerating the DSE process to make early energy-efficient design decisions.

Keywords—Energy-aware, Co-Design, MARTE, MDE, MP2SoC

I. INTRODUCTION

The use of highly integrated System-on-Chip (SoC) to run data intensive multimedia functions has increased rapidly over the past decade. Simultaneously, the semiconductor industry continued to guide technology along the lines of Moore's law making advantage from the gigantic number of transistors that doubled every 1.96 years between 1971 and 2001. Following this historical trend, the only performance concern of complex multimedia applications was the speed of the SoC, which keeps increasing along with the high transistor density. At the Intel Developer Forum, in September 2007, Gordon Moore predicted that his famous law would no longer be valid in ten to fifteen years. The ITRS studied the transistor density variations from 2011 to 2026, and showed that Moore's prediction became a reality since 2013: the rate has slowed to about 1.2 times per year [1]. It was typically accepted, at this stage, that miniaturizing Complementary Metal Oxide Semiconductor (CMOS) circuits, reducing the supply voltage, and increasing the frequency had become impracticable. To

improve system effectiveness, increasing the number of cores in a circuit while limiting core complexity seems more efficient than using a unique complex core. Consequently, Massively Parallel Multi-Processors System-on-Chip (MP2SoC) have become the direction for future scaling and several MP2SoC systems have already been announced. The Intel Xeon-Phi co-processor, for example, contains up to 61 X86 cores, providing 1.2 teraflops of performance. The world's fastest supercomputer according to the TOP500 lists for June 2015 [2], Tianhe-2, includes a total of 3,120,000 cores of both Intel Xeon processors and Intel Xeon Phi co-processors.

As the speed metric of MP2SoCs has increased over time, another metric has become more important: power consumption. Tianhe-2, for example, requires 17.8 kW of power to operate 33.8 trillion calculations per second. Over the past years, the particular focus on speed, which has been the synonym of performance, has led to the emergence of massively parallel systems that consume high amounts of power and produce a large amount of heat.

Power and energy efficiency must now be added to the performance metrics of embedded systems, making performance per watt the new metric of merit. Consequently, power consumption becomes a key criterion to take into consideration during design space exploration [3]. Finding a tradeoff between power consumption and performance early in the design flow in order to satisfy time-to-market is the design challenge of Electronic Design Automation (EDA) tools.

In the recent years, numerous techniques have been integrated into system-level EDA tools to minimize the power consumption in embedded systems. The research challenges tackled by this paper are: (a) proposing a power estimation and optimization approach that takes the consumption criterion into account early in the design flow while achieving a better tradeoff between estimation accuracy and speed (b) integrating a power management technique that considers the power consumption of both processors and interconnects of a given MP2SoC.

The key contribution of the work presented in this paper is the implementation of a scheduling kernel that contains a state-of-the-art power-aware scheduling algorithm: the Energy-Aware Duplication (EAD) algorithm [4]. The scheduling algorithm uses a task duplication strategy to eliminate commu-

nication delay among processors, reducing the overall communication overheads in MP2SoC while saving energy. The scheduling kernel is integrated into an MDE-based Design Space Exploration (DSE) approach to optimize both speed and energy efficiency in MP2SoC. Moreover, the proposed framework extends the Modeling and Analysis of Real-Time and Embedded systems (MARTE) profile with power aspects of MP2SoC systems providing a time-saving specification methodology.

This paper is organized as follows: in the next Section, a literature overview will be highlighted. In Section III, main features of our proposed power-aware DSE methodology are briefly described. Section IV details the introduced MARTE extensions for the specification of power objectives. In Section V, the energy-aware scheduling kernel is detailed. The effectiveness of the approach is demonstrated using the H.263 encoding application as a case study in Section VI.

II. LITERATURE OVERVIEW

In energy-aware EDA tools, the power estimation process is affected by three aspects: the power specification language, the abstraction level of the specification and the available power estimation and optimization techniques.

A. Languages for power specification

There are several studies proposed in the literature aiming to characterize power consumption in embedded systems at different levels of abstraction using specification languages. In an attempt to achieve high accuracy, two languages have emerged to describe power concepts at register transfer level (RTL). The Unified Power Format (UPF) [5] and the Common Power Format (CPF)[6] IEEE standards improve the design, verification and implementation of complex integrated circuits while providing concepts to annotate power supplies and power control of a given design. As we move up to higher levels, SystemC-based power modeling approaches capturing power design characteristics in Transaction-Level Modeling (TLM) have emerged to provide fast estimations and simulations. Authors in [7] extend the CPF/UPF standards with TLM directives to define a system-level power model. Then, the TLM simulation front-end processes an automatic TLM instrumentation process and enables voltage-tuned simulation. Nowadays, the increase of design abstraction levels that Unified Modeling Language (UML) profiles provide, make early power estimation and optimization possible while using UML annotations. SysML and MARTE profiles provide annotations to describe some aspects related to power consumption in embedded systems. To support the modeling of dynamic power management, authors in [8] propose a MARTE extension that relies on UML finite state machines. Another MARTE-based power consumption profile is described in [9]. Authors propose an off-line Dynamic voltage scaling (DVS)-based scheduling algorithm to analyze the power consumption of real-time embedded systems. These extensions are not sufficient for our approach as they only focus on MPSoC systems with a limited number of Processing Elements (PEs). In addition, the energy consumption of NoCs is neglected and the proposed power management techniques are limited to processors.

B. EDA tools for power estimation and optimization

A new research trend is raising that aims at developing EDA tools for power consumption at different abstraction levels moving from RTL level to System level to finally achieve model abstraction level. Among the power optimization tools operating at the RTL level we can mention PETROL [10]. To deal with the long simulation time, SimplePower [11] and Wattch [12] tools have been developed for power consumption estimation at system-level. While allowing accurate power estimation, simulation time keeps increasing when exploring complex architectures. To meet performance requirements and to achieve quick exploration times, the EDA industry relies on MDE approaches demanding for system power consumption estimation at early stage in the design flow. STORM [13], Gaspard2 [14] [15], PETS [16], CAT [17] and TTool [18] are MDE-based power-aware tools that rely on high-level models. While STORM and CAT use AADL-based design entries for system-level power and energy consumption estimation, PETS benefits from the generated SystemC code to estimate the power consumption during simulations. Similar to GASPARD2, which uses the MARTE profile for power specification, the TTool DSE toolkit integrates power concepts in its DIPLODOCUS UML profile. MDE-based methodologies for the power estimation of MPSoC systems defined in [14] and [15] were integrated in the Gaspard2 framework. In [14], the proposed methodology allows one to automatically generate system descriptions at Cycle-Accurate Bit-Accurate (CABA) and Programmers View with Timing (PVT) simulation levels. The same approach was adopted in [15]. The generated simulated architectures in [14] and [15] are used to estimate power consumption. Comparing these related works with our approach, we can observe that none of them uses energy-aware scheduling algorithms for the high-level design space exploration of MP2SoC systems. Moreover, these approaches mainly try to exploit low-level simulations for power analysis. On the contrary, our approach is based on a data-flow based specification for the high-level analysis of MP2SoC.

III. CONTEXT

A. Previous work and limitations

An automatic DSE approach that takes advantage from MDE and MARTE was proposed in [19] [20]. It defines two levels of abstraction that alleviate the analysis and generation of data-intensive processing applications running on MP2SoC architectures (Figure. 1). The first level is based on a novel extension of the famous Synchronous Data Flow (SDF) [21] Model-of-Computation (MoC), the Parameterized and Interfaced Synchronous Dataflow (π SDF) [22] model. Another level is introduced in our platform-based co-design flow facilitating IP integration, architecture generation and system analysis. This level complies with a model based on the IP-XACT standard [23] named System-Level Architecture Model (S-LAM) [24]. High-level MARTE-based specification of the parallel architecture can be then refined in an MDE-based process to produce S-LAM description of the platform. In [19], the UML/MARTE methodology for modeling the data-parallel application and the automatic generation of the π SDF specification have been presented. In [20], the automatic generation from the UML/MARTE specification of the S-LAM description of the architecture was explained. The final step in

the proposed approach is the rapid prototyping of the π SDF/S-LAM/Scenario combination using PREESM [25]. The flexible rapid prototyping process in PREESM consists of exploring the design tradeoffs at system-level while taking into account system constraints and objectives present in a scenario file. The central feature of the rapid prototyping method is the multi-core scheduler. Before starting the scheduling phase, PREESM performs three transformations aiming to expose the parallelism of the application: the π SDF graph is transformed into a Hierarchical SDF, then into a single rate SDF and finally into a DAG. The latter is processed by the proposed scheduler. Prototyping complex application using the scheduling kernel of PREESM brings some limitations including:

- Lack of energy estimation and optimization
- Scheduling with a bounded set of processors

In fact, performance is evaluated based on two metrics, throughput and latency. At the end of the scheduling process, a Gantt chart of the execution is displayed, plotting the optimal schedule. Memory storage requirements and speedup values are also estimated and plotted in different charts. Although the optimization of these constraints is vital when dealing with high-performance applications, limited power consumption is becoming an even more important objective with the ever increasing number of cores inside MP2SoC systems.

In addition, the static scheduling algorithms implemented within the PREESM scheduler, including the list scheduling and the FAST algorithms, are mainly dedicated to scheduling tasks on MP2SoC systems with a bounded number of processors.

B. Energy optimization and performance estimation framework

Task partitioning and scheduling approaches take important part in achieving high performance for parallel applications on MP2SoC systems.

Recently, many State-of-the-Art studies dealing with power-aware scheduling have been conducted, demonstrating that Dynamic Voltage and Frequency Scaling (DVFS) technique is one of the most efficient strategies to reduce energy consumption in power-scalable MP2SoCs. The Massively Parallel Processor Array (MPPA-256) [26], for example, implements the DVFS power management technique to achieve 75 GFLOPS/W of energy efficiency. MPPA-256 has an array of 16 clusters connected through a high-speed NoC with a bandwidth up to 3.2 GB/s.

While DVFS has taken part in designing energy-efficient MP2SoCs, most of them are only capable of saving energy in processors executing computation-intensive applications. As a result, the benefits of DVFS may diminish when it comes to communication-intensive applications, because the energy consumed by interconnects dominates the total power consumption and energy saving techniques for MP2SoC interconnects do not exist [27]. This situation is getting worse with the emergence of complex massively parallel NoCs that guarantee high-speed while consuming more energy. In addition, some embedded processors do not support the DVFS technique, making impossible to vary voltage and frequency of the MP2SoC processors to decrease the energy consumption. The rising static power

consumption and reduced dynamic power consumption of next-generation processors, are also diminishing the benefits of DVFS [28].

Duplication-based scheduling has proven to be an efficient strategy [29] to schedule parallel tasks while minimizing communication overhead. Emerging duplication-based approaches struggle to minimize schedule lengths at the cost of energy consumption. Researches in this field try to combine duplication-based algorithms with power reduction [29]. These efforts use emerging power reduction techniques and try to adapt them for cluster-based systems.

Following this direction, we studied a power-aware duplication-based scheduling algorithm, EAD, proposed in the context of homogenous cluster-based systems [4]. We conclude that integrating such technique into the proposed framework is a promising direction since we target homogenous MP2SoC systems containing one cluster of processing units. Another motivating point is that state-of-the-art techniques are based on a DAG description of the application [30], which is the entry point of the PREESM scheduler.

Integrating power estimation and optimization concepts in the proposed framework follows four major steps:

- Adding power annotation capabilities to the MARTE profile
- Integrating the needed power information in the framework meta-models (MARTE and S-LAM) in order to automate the estimation and optimization process,
- Using timing and power information gathered from S-LAM model, scenario file, and π SDF model, to generate a timed DAG,
- Performing energy estimation and optimisation using the scheduling kernel that contains the energy-aware duplication-based algorithm.

The next Section will detail the first step.

IV. INTRODUCED EXTENSIONS FOR POWER MODELING

The sources of power consumption of MP2SoC components are dynamic power and static power as given by equation:

$$P = P^{dyn} + P^{stat} \quad (1)$$

The interconnection network is characterized with the corresponding static power consumption and dynamic power consumption. The dynamic power consumption of a processing element is in turn dependent on a set of parameters as follows:

$$P_{PE}^{dyn} = e_{cycle} \cdot \alpha \cdot f \quad (2)$$

where e_{cycle} is the maximum energy per clock cycle, α is the switching activity factor, and f is the operating frequency of the processing element. These parameters should be defined in the MARTE profile in order to enable power-aware scheduling.

MARTE proposes a power sub-package (*HW_Power*) in its Hardware Resource Modeling (*HRM*) package, where power consumption of each hardware component can be specified. In addition, it allows annotating non-functional properties

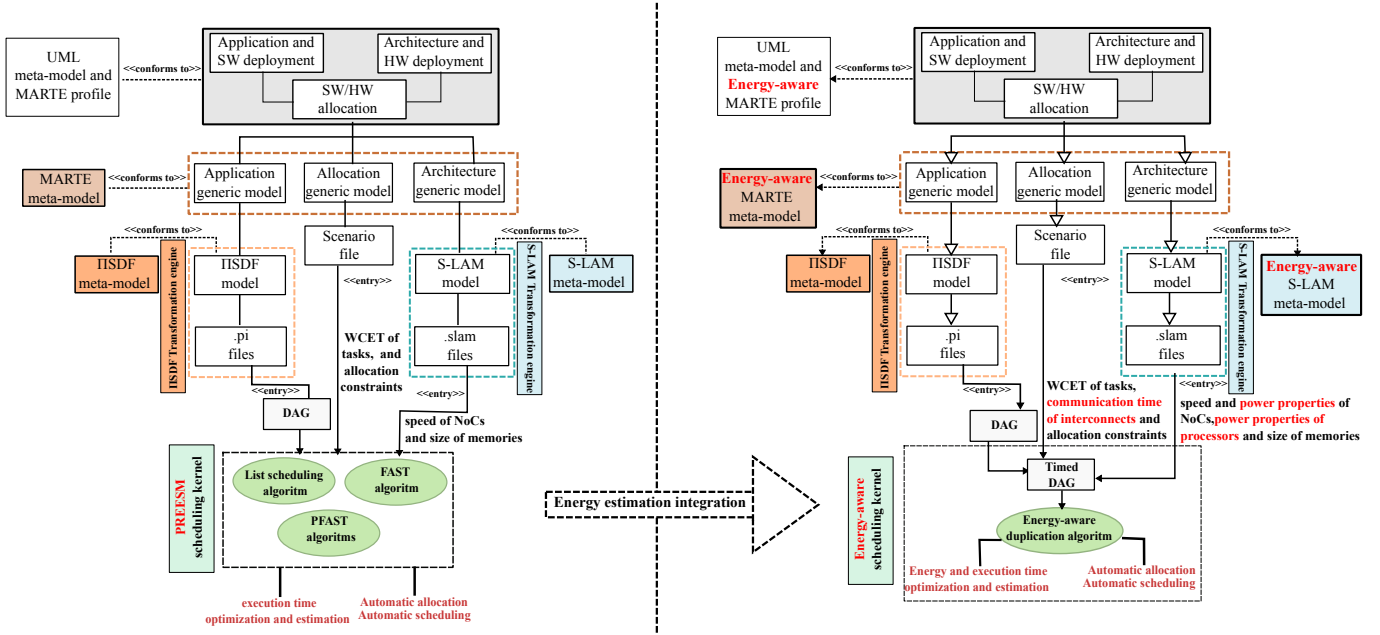


Fig. 1. Proposed energy-aware DSE flow

related to power and energy using power-related attributes from the *HwPowerSupply* or the *HwComponent* stereotypes.

The main idea of our specification methodology is that each hardware component is associated with the appropriate stereotype from the *HW_Logical* package defining its functional properties (*HwProcessor*, *HwMemory*, *HwCommunicationResource*). Moreover, each processing element and each interconnect is annotated with the *HwComponent* stereotype. This stereotype presents each hardware resource as a physical component with details on its physical properties including power characteristics.

To provide accurate estimation adopting the selected energy consumption model, additional power-related expressions are needed. In fact, the *HwComponent* stereotype provides specification of static power consumption specification using the *staticConsumption* attribute. While the static power consumption of a given component can be annotated, MARTE disregards the dynamic consumption associated with the component activity. Consequently, the power of PEs and the MP2SoC interconnect in busy working mode cannot be modeled. Figure. 2 illustrates the *HwComponent* stereotype enriched with other attributes for high-level dynamic power modeling. *energyPerCycle*, *switchingActivity*, and *frequency* attributes can feed a computation of the dynamic power consumption of a given processing element using Equation (2). The *dynamicConsumption* attribute expresses the average dynamic power consumption of an interconnection network. This attribute can be also needed in case there is no available information about the energy per cycle, the switching activity, or the frequency of a given processing element.

V. ENERGY-AWARE SCHEDULING KERNEL

Increasing concurrency, while decreasing inter-processor communication cost, is a key challenge when scheduling a DAG on a multiprocessor architecture. Therefore, finding an

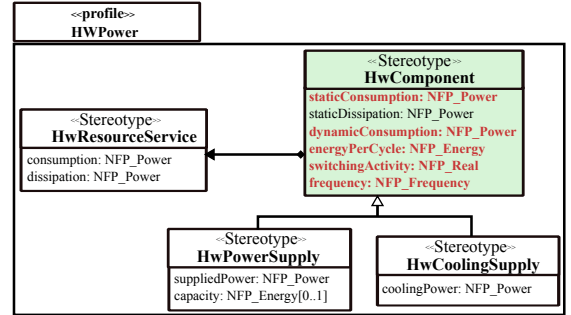


Fig. 2. Proposed MARTE extensions

optimal schedule is a NP-hard problem [31]. A method to decrease inter-processor communication cost is task duplication-based scheduling.

The central idea behind duplicating tasks is to benefit from processor idling time to remove waiting periods on other processors by duplicating predecessor tasks. This technique prevents transfer of results via the communication network from a predecessor. To our knowledge, this is the first time that an energy-aware duplication scheduling algorithm dedicated to cluster environments is integrated in a model-based co-design framework. The duplication process of the EAD algorithm is similar to those found in other state-of-the-art duplication-based scheduling schemes.

The EAD algorithm runs in three steps:

In the first step, the DAG is navigated in a top-down fashion to compute the level for each node and create a task sequence. The elements in the task sequence are the tasks sorted in the ascending order of level.

In the second step, important parameters for each task are computed. Mathematical equations used to calculate these

parameters can be found in [4].

In the third step, the EAD algorithm will make task duplication decisions while guaranteeing optimal energy consumption. In fact, it groups communication-intensive parallel tasks and allocates them to the same processing element. Moreover, it makes trade-offs between schedule lengths and energy savings using an energy consumption model.

The proposed energy model in the EAD algorithm was modified to be compatible with the characteristics of MP2SoC systems.

The architectures targeted by our framework are distributed memory MP2SoC systems containing more than one hundred homogeneous PEs connected via a fast network. These architectures are composed of an SIMD cluster. The cluster includes a configurable number of identical PEs.

A homogeneous SIMD cluster is defined as a set $PE = \{PE_1, PE_2, \dots, PE_n\}$, where PE_i is a processing element attached to its local memory.

For making explicit duplication choices inside the energy-aware kernel, refinements should be performed to produce a timed DAG description of the application as explained in Section III.

A timed DAG is a directed graph $G = (V, E)$ where:

- $V = \{v_1, v_2, v_N\}$ is the vertex set of tasks, with t_i is the execution time of v_i and $1 \leq i \leq N$
- E is the edge set, with $e_{ij} = (v_i, v_j, c_{ij})$ a message communicated between tasks v_i and v_j having a communication time c_{ij}

The total energy consumed when running a parallel application on an MP2SoC system is estimated using Equation (3) where $E_{PEclust}$ presents the total energy consumption of the PE cluster and E_{NoC} depicts the energy consumption of the entire interconnection network.

$$E = E_{PEclust} + E_{NoC} \quad (3)$$

The average energy consumption in digital circuits consists of two main components: dynamic energy and static energy. Therefore, the overall energy consumption of the PE cluster and the interconnection network can be defined as the summation of dynamic and static energy consumption as seen in Equation (4) and (5).

$$E_{PEclust} = E_{PEclust}^{dyn} + E_{PEclust}^{stat} \quad (4)$$

$$E_{NoC} = E_{NoC}^{dyn} + E_{NoC}^{stat} \quad (5)$$

Equation (6), (7), (8), and (9) give the detailed energy estimation model integrated in the proposed framework.

$$E_{PEclust}^{dyn} = P_{PE}^{dyn} \sum_{i=1}^n t_i^{busy} = (e_{cycle} \cdot \alpha \cdot f) \sum_{i=1}^n t_i^{busy} \quad (6)$$

$$E_{PEclust}^{stat} = P_{PE}^{stat} \sum_{i=1}^n t_i^{idle} \quad (7)$$

$$E_{NoC}^{dyn} = P_{NoC}^{dyn} \sum_{i=1}^n \sum_{j=1, j \neq i}^n c_{ij}^{busy} \quad (8)$$

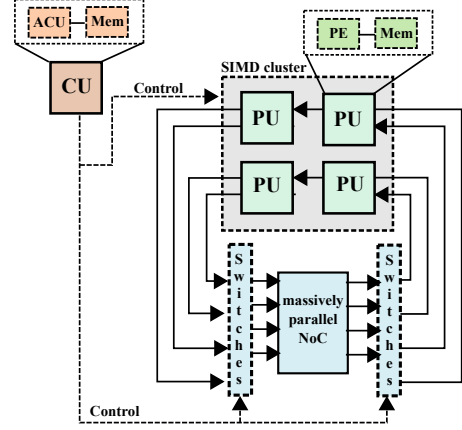


Fig. 3. The SIMD MP2SoC system

$$E_{NoC}^{stat} = P_{NoC}^{stat} \sum_{i=1}^n \sum_{j=1, j \neq i}^n c_{ij}^{idle} \quad (9)$$

VI. CASE STUDY: AN H.263 ENCODER

In this study, we chose to use the H.263 video codec, a mature and popular coding standard [32]. This application is taken from the *SDF*³ Benchmark [33] with worst-case execution times for an ARM7TDMI core.

A. Simulation parameters

1) *Hardware simulation parameters*: The experimental platform, shown in Figure. 3, is an SIMD massively parallel processing SoC composed of a parametric set of PEs [34]. The SIMD cluster encloses homogenous ARM7TDMI cores, with private and local data memories attached to each core. The size of each local memory is parametric and can be configured depending on the application storage needs. To satisfy the requirements of complex applications, the platform contains a massively parallel crossbar-based NoC reaching 30MB/s of bit-rate. It is a flexible and reconfigurable network performing point to point irregular communications. In fact, the interconnect interface of the NoC is generic enough to support a configurable size of inputs and outputs which are equal to the number of PEs in the SIMD cluster. The SIMD cluster and the massively parallel NoC are controlled synchronously by an Array Controller Unit (ACU) which is responsible of transferring parallel instructions to the cluster and handling control or serial computations. The power consumption rates of the ARM7TDMI cores [35] and the massively parallel NoC used in the system specification are summarized in Table I.

2) *Software simulation parameters*: The basic coding architecture of H.263 encloses an encoder part and a decoder part [32]. Several application parameters can be adjusted and optimized to meet time and power constraints. For instance, data-parallelism can be exploited to reduce the execution time of the application by taking advantage of the SIMD massively parallel structure of the cluster. In H.263, data parallelism at macro-block (MB) level permits to execute tasks of the codec on different group of macroblocks (GOMB) in parallel. To study the tradeoff between parallelism and energy, the macro-block level parallelism is exploited in the

TABLE I. HARDWARE AND SOFTWARE SIMULATION PARAMETERS SUMMARY

Software	Tested frames	Resolution	SQCIF	QCIF
		Size in pixel	128*96	176*144
		Size in MB	48	99
		GOMB	4, 8, 16, 48	3, 9, 11, 99
Hardware	Processor	Name	ARM7 TDMI	
		Frequency	100 MHz	
		Energy per cycle	0.39 mW / MHz	
		Static power	16mW	
	NoC	Switching activity	1	
		Bitrate	30 MB / s	
		Static power	15 mW	
		Dynamic power	20 mW	

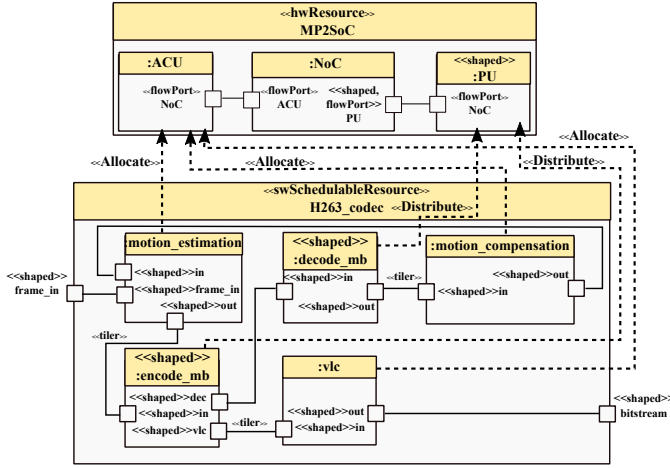


Fig. 4. Application, architecture and allocation UML diagram

experiments on two widely known image resolutions, SQCIF and QCIF, varying the number of macro-blocks processed in parallel as seen in Table I. For each simulation, the execution time of tasks in the H.263 application from the *SDF*³ Benchmark [33] are defined in the UML model of the application using the *deadlineElements* attribute from the *swSchedulableResource* stereotype.

B. Experimental results

To rapidly design an MP2SoC system that meets its constraints, in particular those related to timing and energy, two main steps are identified: high level system specification and system-level analyses.

1) *High-level system specification*: The H.263 codec UML model sketched in Figure. 4 models the application functionality. The targeted architecture is composed of a parametric set of processing units (PU), containing each a processing element connected to its local memory, an ACU, and a shared NoC, as illustrated in Figure. 4. The mapping of the application onto the MP2SoC architecture is sketched in the same figure. The sequential tasks of the H263 codec are mapped on the ACU via the *Allocate* links specified in Figure. 4. The *Distribute* stereotype specifies precisely the distribution of the repetitions of encode_mb and decode_mb tasks onto the SIMD cluster containing the parametric set of PUs. The parametric specification allows the scheduler taking partitioning and scheduling decisions without limiting the PU number.

2) *Successive transformations*: Once the UML/MARTE-based models are specified, the second step of our energy-

TABLE II. CHARACTERISTICS OF THE GENERATED DAGS

Resolution	GOMB number	Generated DAG	
		Actors number	FIFOs number
SQCIF (128*96) 48MB	4	144	19
	8	22	35
	16	38	67
	48	102	195
QCIF (176*144) 99MB	3	12	15
	9	24	39
	11	28	47
	99	204	399

aware methodology is performed. It involves successive model transformations and system-level analyses of the MP2SoC system. The π sdf transformation chain leads to the generation of a π SDF graph. The S-LAM transformation engine produces an S-LAM description of the SIMD MP2SoC architecture containing the physical properties of the architecture, such as the energy consumption of the PEs and the NoC and the speed of the NoC. The proposed co-design methodology encloses a scenario-based design space exploration that exploits the scenario file generated from high-level model to evaluate a single design point. This means that during the analysis step of the H.263 codec, 8 scenarios are generated and processed separately. Each scenario includes different execution time and communication time values. Moreover, the size of the frame and the number of processed MBs varies from one scenario to another. For each scenario, the user returns to the specification step, change the appropriate parameters, and re-executes the transformations. While values in the scenario file are re-generated for each scenario, the π SDF and the S-LAM files remain the same, permitting a time-saving in the exploration process, which justifies the separation of concerns in the analysis step. To run the energy-aware exploration process on the scenario set, we take advantage of the facilities provided by the PREESM framework. The input models of PREESM (π SDF graph, S-LAM diagram, scenario file) are first obtained, then, the graph transformations module of PREESM is used to convert the generated π SDF model into a DAG before being transformed into a timed DAG and scheduled using the proposed energy-aware scheduling kernel. For each resolution, four DAGs are generated using the PREESM transformation module with different characteristics (number of actors and FIFOs) as seen in Table II.

3) *Executing EAD*: The scheduling kernel estimates the optimal allocation/scheduling schema while choosing the adequate number of PEs as seen in Table III. Figure. 5 illustrates the generated schedule of the DAG containing 14 actors and 19 FIFOs after and before duplicating. The proposed schedule before duplicating reduces the schedule length by allowing encode_mb and decode_mb tasks running in parallel on four computing nodes. The duplication schedule further improves the performance by duplicating motion_estimation and explode tasks on the second, third, and fourth nodes. Thus, the communication delays between the explode task and the encode_mb tasks are eliminated. After duplicating, the communication energy cost decreases from 115536 nJ to 39496 nJ, achieving 65% of gain. The scheduled length is also decreased by a factor of 13%.

One can notice that the H.263 encoding energy and power consumptions depend on the number of processing units and the frame resolution as seen in Figure. 6. The energy and power

TABLE III. GENERATED NUMBER OF PES AND EAD EXECUTION TIME

Resolution	GOMB number	Estimated number of PEs	EAD time (ms)
SQCIF (128*96) 48MB	4	5	59
	8	9	121
	16	17	124
	48	96	380
QCIF (176*144) 99MB	3	4	94
	9	10	112
	11	12	126
	99	198	1241

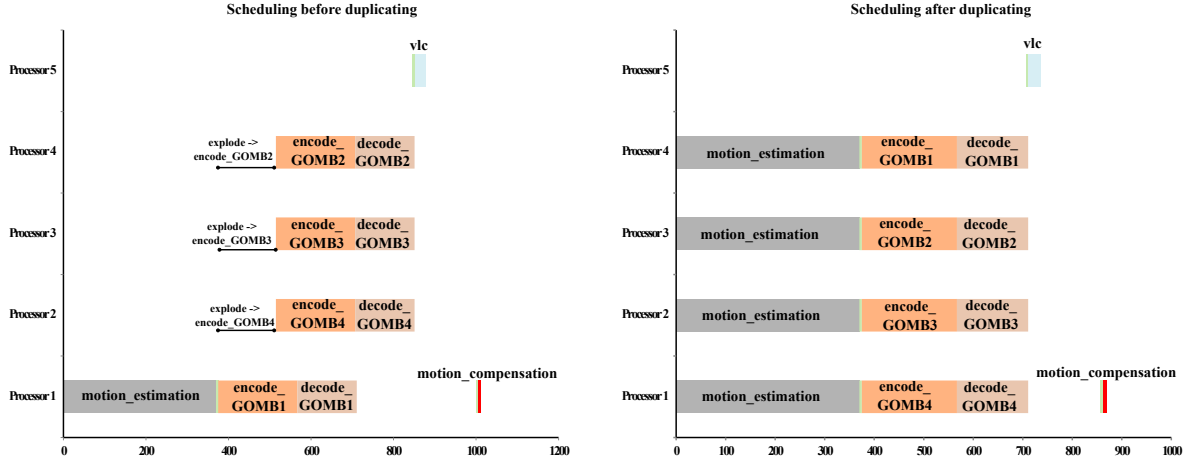


Fig. 5. H.263 encoding 4 GOMBs DAG scheduling

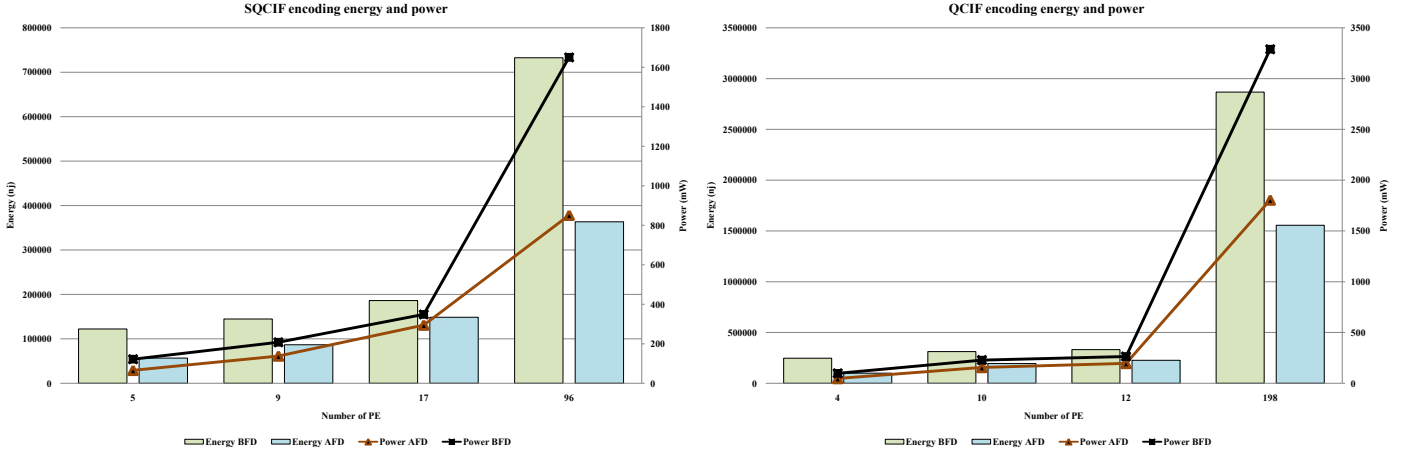


Fig. 6. H.263 encoding energy and power consumption variations

consumptions of a frame increase for high number of PEs. One can observe that the energy consumption variation of the SQCIF encoding differs from that of the QCIF encoding. In fact, the SQCIF consumes less energy than the QCIF since it contains less MBs. One can also infer that for the same resolution, energy measured before duplication (BFD) is bigger than energy measured after duplication (AFD), the fact that demonstrates the effectiveness of the scheduling policy. In fact, the energy gain reached 53% for the SQCIF encoding and 59% for the QCIF encoding. Moreover, gain increasing is directly related to the communication-computation ratio: the more the application is communication-intensive; the more the energy gain is proven.

The obtained results demonstrated that EAD can effec-

tively save energy in MP2SoC systems and keeps respectable speedup. In addition, the proposed scheduling kernel accelerates the DSE process to make early energy-efficient design decisions. The total time required by EAD to make scheduling decisions evaluates the time-efficiency of the proposed DSE flow. EAD time efficiency means time complexity. The time complexity of EAD is $O(2|E| + |V|(\log|V| + 1) + h|V|)$ [4], where E is the number of messages, V is the number of parallel tasks, and h is the height of the DAG. This time complexity demonstrates that even with increased size of DAGs, the exploration time keeps negligible as shown in Table III.

VII. CONCLUSION

This paper proposes an estimation and optimization framework for static power analysis for MP2SoC systems at model-level. To our knowledge, it is the first tool to integrate energy-aware duplication-based scheduling algorithms for the state-of-the-art power-aware MDE-based tools. First, a power modeling methodology has been proposed as an extension to the MARTE profile, to address the global system consumption that includes homogenous PEs and high-speed NoC. Secondly, the studied Energy-Aware Duplication algorithm is coupled with the successive MDE transformations to get the information necessitated by for the scheduling kernel with a better trade-off between accuracy and speed. Experimental results show that our framework can reach important energy gains while facilitating and accelerating the exploration of several implementation choices.

REFERENCES

- [1] MPU (High-volume Microprocessor) Cost-Performance Product Generations and Chip Size Model, INTERNATIONAL TECHNOLOGY ROADMAP FOR SEMICONDUCTORS, 2012.
- [2] TOP 500, <http://www.top500.org/>, 2015.
- [3] M. Grant, "Overview of the MPSoC design challenge," Proceedings of the 43rd annual Design Automation Conference, 2006.
- [4] Z. Zong, et al, "EAD and PEBD: two energy-aware duplication scheduling algorithms for parallel tasks on homogeneous clusters," IEEE Transactions on Computers, vol. 60, no. 3, pp. 360-374, 2011.
- [5] Unified Power Format (UPF 2.0) Standard, "IEEE standard for design and verification of low power integrated circuits," IEEE 1801TM, March, 2009.
- [6] Si2 Common Power Format Specification (CPF 2.1), <http://www.si2.org/?page=811>, 2015.
- [7] F. Mischkalla, and W. Mueller, "Architectural low-power design using transaction-based system modeling and simulation," In 2014 International Conference on Embedded Computer Systems: Architectures, Modeling, and Simulation, SAMOS XIV, pp. 258-265, July 2014.
- [8] T. Arpinen, E. Salminen, T. D. Hämäläinen, and M. Hännikäinen, "MARTE profile extension for modeling dynamic power management of embedded systems," Journal of Systems Architecture, vol. 58, no 5, p. 209-219, 2012.
- [9] M. Hagner, A. Aniculaesei, and U. Goltz, "UML-based analysis of power consumption for real-time embedded systems," IEEE 10th International Conference on Trust, Security and Privacy in Computing and Communications, TrustCom 2011, pp. 1196-1201, November 2011, .
- [10] R. Peset-Lopis and K. Goossens, "The petrol approach to high-level power estimation," Proceedings of the ISLPED, Monterey, California, USA, august 1998.
- [11] W. Ye, N. Vijaykrishnan, M. Kandemir, and M. Irwin, "The design and use of SimplePower: A cycle accurate energy estimation tool," In Proceedings of the 37th Annual Design Automation Conference, pp. 340-345, June 2000.
- [12] D. Brooks, V. Tiwari, and M. Martonosi, "Wattch : A framework for architectural-level power analysis and optimizations," vol. 28, no. 2, ACM, 2000.
- [13] Storm simulation tool, <http://storm.rts-software.org>, 2015.
- [14] R. Ben Atitallah, E. Piel, S. Niar, P. Marquet, and J.-L. Dekeyser, "Multilevel MPSoC simulation using an MDE approach," In Proceedings of the IEEE International SOC Conference, SOCC'07, pp.197-200, Hsin Chu, Taiwan, September 2007.
- [15] C. Trabelsi, R. Ben Atitallah, S. Meftali, J.-L. Dekeyser, and A. Jemai, "A model-driven approach for hybrid power estimation in embedded systems designs," In EURASIP Journal on Embedded Systems, 2011.
- [16] S.-K. Rethinagiri, O. Palomar, O. Unsal, A. Cristal, R. Ben Atitallah, and S. Niar, "Pets: Power and energy estimation tool at system-level," IEEE 15th International Symposium on Quality Electronic Design, ISQED, pp. 535-542, 2014.
- [17] E. Senn, S. Douhib, D. Blouin, J. Laurent, S. Turki, and J.-P. Diguët, "Power and Energy Estimations in Model-Based Design," In Languages for Embedded Systems and Their Applications, pp. 3-26, Springer Netherlands, 2009.
- [18] F. Ben Abdallah, C. Trabelsi, R. Ben Atitallah, and M. Abed, "Early power-aware Design Space Exploration for embedded systems: MPEG-2 case study," In 2014 International Symposium on System-on-Chip, SoC, pp. 1-8, October 2014.
- [19] M. Ammar, M. Baklouti, M. Pelcat, K. Desnos and M. Abid, "MARTE to π SDF transformation for data-intensive applications analysis," In Conference on Design & Architectures for Signal & Image Processing, DASIP, October 2014.
- [20] M. Ammar, M. Baklouti, M. Pelcat, K. Desnos and M. Abid, "Automatic Generation of S-LAM Descriptions from UML/MARTE for the DSE of Massively Parallel Embedded Systems," In Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, pp. 195-211, 2015, Springer International Publishing.
- [21] E. Lee and D. Messerschmitt, "Synchronous data flow," Proceedings of the IEEE, vol. 75, no. 9, pp. 1235-1245, September 1987.
- [22] K. Desnos, M. Pelcat, J.F. Nezan, S. Bhattacharyya and S. Aridhi, "PiMM: Parameterized and Interfaced Dataflow Meta-Model for MP-SoCs Runtime Reconfiguration," International Conference on Embedded Computer Systems: Architecture, Modeling and Simulation, SAMOS XIII, Greece, July 2013.
- [23] IEEE Standard for IP-XACT, Standard Structure for Packaging, Integrating, and Reusing IP within Tools Flows, IEEE Std 1685-2009, Feb. 2010, pp. C1-360.
- [24] M. Pelcat, J. F. Nezan, J. Piat, J. Croizer, and S. Aridhi, "A system-level architecture model for rapid prototyping of heterogeneous multicore embedded systems," In Conference on Design & Architectures for Signal & Image Processing, September 2009.
- [25] M. Pelcat, K. Desnos, J. Heulot, C. Guy, J.-F. Nezan, and S. Aridhi, "Preesm: A dataflow-based rapid prototyping framework for simplifying multicore DSP programming," In 6th European Embedded Design in Education and Research Conference, EDERC 2014, pp. 36-40, 2014.
- [26] B.D. de Dinechin, et al, "A clustered manycore processor architecture for embedded and accelerated applications," 2013 IEEE High Performance Extreme Computing Conference, HPEC, 2013.
- [27] M. Etinski et al, "Understanding the future of energy-performance trade-off via DVFS in HPC environments," Journal of Parallel and Distributed Computing, vol. 72, no 4, pp. 579-590, 2012.
- [28] E. Le Sueur, and G. Heiser, "Dynamic voltage and frequency scaling: The laws of diminishing returns," Proceedings of the 2010 international conference on Power aware computing and systems, USENIX Association, 2010.
- [29] G. L. Valentini, W. Lassonde, S. U. Khan, et al, "An overview of energy efficiency techniques in cluster computing systems," Cluster Computing, vol. 16, no. 1, pp. 3-15, 2013.
- [30] L. WANG, S. U. KHAN, D. CHEN, et al, "Energy-aware parallel task scheduling in a cluster," Future Generation Computer Systems, vol. 29, no. 7, pp. 1661-1670, 2013.
- [31] H. Kasahara, and N. Seinosuke, "Practical multiprocessor scheduling algorithms for efficient parallel processing," IEEE Transactions on Computers, no. 11, pp. 1023-1029, 1984.
- [32] K. Rijkse, "H. 263: video coding for low-bit-rate communication," IEEE Communications Magazine, vol. 34, no. 12, pp. 42-45, 1996.
- [33] S. Stuijk, M. Geilen, and T. Basten, "SDF3: SDF For Free," In Proceeding Application of Concurrency to System Design, pp. 276-278, 2006.
- [34] M. Baklouti, P. Marquet, J. -L. Dekeyser, and M. Abid, "FPGA-based many-core System-on-Chip design," Microprocessors and Microsystems, 2015.
- [35] S. Segars, "ARM7TDMI power consumption," Micro, IEEE, vol. 17, no. 4, pp. 12-19, 1997.